

PHONETIC-BASED TEXT INPUT METHOD

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[01] The present invention relates to the transliteration of user input into a computing device. More specifically, the invention provides methods for inputting language into a computing device based on a phonetic-based scheme.

DESCRIPTION OF RELATED ART

[02] One of the early challenges that faced operating systems developers was how to make operating systems work with many different languages and scripts to allow persons from various different cultures to use the system effectively in their native language.

[03] Early operating systems were typically exclusively text-based and were only able to display characters from the Latin alphabet, i.e., the standard “a b c d ...” alphabet used to write the English and other Western European languages. These early systems were unable to display the complex characters and symbols from scripts and alphabets used to write languages such as Chinese, Japanese, Russian, Arabic, Korean, Hindi, Sanskrit, and various other languages that utilize non-Latin alphabets.. In fact, many of the early text-based operating systems were unable to fully display those Western European languages that utilize special letters and accents (for example, the German umlaut character “ä”).

[04] As GUI-based operating systems such as Microsoft® Windows became preferred over text-based system such as DOS, many realized that graphics-based operating systems made it possible to adapt to other more complex characters from non-Latin scripts and alphabets. However, even with the improved ability to display non-Latin alphabets, a problem surfaced. Developers from different countries failed to use standardized tools to create systems that provided multi-language support. Various

national and regional standards had been developed for displaying characters. These standards included ASCII for American English, JIS for Japanese, GB Code for Chinese (PRC), Big-5 for Chinese (Taiwan), KS-Code for Korean, and ISCII for Indic languages (i.e. languages of India). Unfortunately, these standards were not compatible with one another and as a result, providing multiple language support in these new GUI-based operating systems was nearly impossible.

- [05] In order to rectify this problem, a coalition of engineers from large software companies convened in the late 1980's and proposed a standard known as Unicode. Unicode is a 16-bit code that allows for 2 to the 16th power (65,536) different characters to be defined within the code. Unicode allowed characters/symbols in each language to be assigned a unique value such that when the computer system reads that value, it knows exactly which character/symbol to display on the screen.
- [06] Soon thereafter, Unicode became the accepted standard for rendering characters and symbols in computer operating systems, and it became possible for a user to type and display characters from many different languages within the same computer system. Once it was possible to display characters from the various world languages, development efforts began to focus on how best to provide users with an interface for entering characters and symbols from the various languages into the system.
- [07] One solution was the development of keyboard layouts. A keyboard layout is the collection of data for each keystroke and shift state combination within a keyboard software driver. A keyboard layout is distinguishable from the physical keyboard. The physical keyboard is comprised of the actual keys that a user strikes with his fingers to provide input to the computer. Most keyboards are physically the same, as shown in Figure 2. Each key on the physical keyboard has a value assigned to it called a scan code. A key's scan code is sent to the computer each time that key is struck by the user. The shift state (i.e. whether the shift key is being held down) of the physical keyboard is sent as well. This code and the shift state must be interpreted

by the software to determine what to display on the screen in response to typing that key. Figure 3a shows a scan code mapping for a typical U.S. keyboard.

- [08] The keyboard layout is the software that is used to receive the hardware call (i.e. the scan code from the physical keyboard) and output text based on the scan code and shift state to the active application. Many languages have keyboard layouts that are unique to that language. Each keyboard layout is configured to call a specific Unicode value for each scan code/shift state combination. For example, striking the key with the scan code 0x10 in an unshifted state will cause the US English keyboard layout to display the 'q' character as can be seen in Figure 3b. Striking that same key when using a keyboard layout configured for the Hindi language will result in the ढ character being displayed by the system.
- [09] In the Microsoft® Windows operating system environment, users may switch between input languages by using the language toolbar. Referring to Figure 4, a language bar 400 is shown. The language bar 400 includes a language selection dropdown menu 402 which allows the user to switch between the various installed input languages – for example, English (United States) 404, French (France) 406, or Hindi 408. It may also include other columns that provide the user with other options to configure the input of different languages into the system.
- [10] Utilizing different keyboard layouts to input the different Unicode characters that represent various languages is useful when user knows the particular keyboard layout. However, in some cases, users are only familiar with keyboard layouts in a single language and do not know keyboard layouts for additional languages. Thus, there is a need for an input method allowing input of multiple languages while not requiring a user to learn a new keyboard layout.

BRIEF SUMMARY OF THE INVENTION

- [11] A method is provided for inputting Unicode characters without having to know a keyboard layout that corresponds to the language that is being typed into the system. The method involves using inputted phonetic strings to output Unicode strings that can be displayed by the operating system as equivalent representations of the inputted phonetic strings.
- [12] In one aspect of the invention, a computer implemented method is provided in which phonetic input to an application is received by the computer. Prior to reaching the application, the input is hooked (i.e. captured) by the computer system. The system then converts the hooked phonetic input into a language. The converted input is then passed to the application for display on the computer.
- [13] In another aspect of the invention, a method is provided for transliterating a first language to a second language. The method involves displaying a text string in a first language. The text string is submitted to a phonetic mapping engine that has a phonetic mapping scheme for the first language, where it is converted into a phonetic text string. The phonetic mapping engine then, based on a phonetic mapping scheme for a second language, converts the phonetic string into the second language, and displays the converted string on the computing device.

BRIEF DESCRIPTION OF THE DRAWINGS

- [14] The present invention is illustrated by way of example and not limited in the accompanying figures in which like reference numerals indicate similar elements and in which:
- [15] Figure 1 shows a general purpose computer that provides a suitable environment to practice the present invention.
- [16] Figure 2 shows a standard US English physical keyboard as is known in the art.

- [17] Figure 3 shows a scan code mapping for the US English physical keyboard and keyboard layouts for the English and Hindi languages.
- [18] Figure 4 shows an example of a language bar that is known in the prior art.
- [19] Figure 5 shows a block diagram of the basic logical structure and data flow of the phonetic mapping engine that may be used to implement aspects of the present invention.
- [20] Figure 6 shows an example of a Hindi phonetic mapping scheme.
- [21] Figure 7 shows an example of a Telugu phonetic mapping scheme.
- [22] Figure 8 shows steps used for transliterating characters according to an aspect of the present invention.
- [23] Figure 9 shows an embodiment of a preview mechanism and a Language Bar that may be used to invoke aspects of the present invention.
- [24] Figure 10 shows the logical flow that may be used to transliterate text from one language to another according to an aspect of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

- [25] In the following description of the various embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration various embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural and functional modifications may be made without departing from the scope of the present invention.
- [26] Figure 1 illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should

the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

- [27] The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.
- [28] The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.
- [29] With reference to Figure 1, an exemplary system for implementing the invention includes a general-purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA

(EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[30] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[31] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such

as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Figure 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[32] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Figure 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[33] The drives and their associated computer storage media discussed above and illustrated in Figure 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Figure 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum,

they are different copies. A user may enter commands and information into the computer 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

- [34] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.
- [35] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked

environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Figure 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

- [36] The present invention overcomes limitations and deficiencies found in the prior art by using a phonetic mapping engine to provide transliteration of text inputted into a computer system. Transliteration refers to the expression of words or letters of a language using characters of a non-native alphabet to that language. The phonetic mapping engine receives an input string in a first alphabet, and based on mapping schemes known to the phonetic mapping engine, maps the input string to a Unicode character output string in a second alphabet.
- [37] Referring to Figure 5, a block diagram shows the logical structure of a phonetic mapping engine 502 along with the data flow in and out of the engine. The phonetic mapping engine may be implemented in software stored in system memory 130, in hard disk 141, or on some other storage media known in the art. It may be part of operating system 134,144 or it may be a separate application program 135,145. The phonetic mapping engine 502 uses phonetic mapping schemes 504 that provide mappings of phonetic strings to Unicode characters. Phonetic mapping schemes 504 may provide mappings of phonetic strings received in one alphabet to the alphabet of a given or desired language. The phonetic mapping engine 502 receives an input string 506 which, in an illustrative embodiment, may be a phonetic string in the form of text input as English characters. The phonetic mapping engine 502 receives the input string 506, and converts it to a Unicode character string based on a phonetic mapping scheme 504 that has been provided to the phonetic mapping engine 502. Once the conversion has been completed, the phonetic mapping engine 502 produces

Unicode output 508 which is an equivalent representation (in a Unicode supported language) for the phonetic string given.

- [38] To illustrate a phonetic mapping scheme 504, examples of two phonetic mappings schemes are provided in Figure 6 and Figure 7. Figure 6 provides an illustrative phonetic mapping scheme 600 for the Hindi language and alphabet. The Hindi language is a language widely spoken and written in India. This Hindi mapping scheme 600 maps phonetic strings 602 typed in the English US alphabet to corresponding Unicode strings 604 that represent the Hindi equivalent of the phonetic string. It may be appreciated by one of skill in the art that the phonetic strings 602 from which the Hindi equivalents are mapped need not necessarily be limited to the English US alphabet. Figure 7 provides an illustrative phonetic mapping scheme for the Telugu language 700 (also primarily spoken in India). The phonetic strings 702 on the left side in the phonetic mapping scheme 700 for the Telugu language are also typed in the English US alphabet and mapped to corresponding Unicode strings 704 that represent Telugu equivalents. Those of skill in the art will appreciate that a mapping scheme to map from any first alphabet to any language that does not natively use that alphabet may be used.
- [39] In an embodiment of the present invention, a method is provided for inputting a language into a computing device using a phonetic mapping engine as described above. This embodiment of the present invention may be used when a user wishes to type in a certain language, but is unfamiliar with the keyboard layout that is typically associated with that language. For example, an English speaking user may wish to type a letter in Hindi, but does not know the Hindi keyboard. The user, however, is familiar with the US English keyboard. The user, presumably familiar with the phonetic mapping scheme for the Hindi language 600 (shown in detail in Figure 6), e.g., a bi-lingual English/Hindi speaker, may, using the US English keyboard layout, type phonetic strings (according to the Hindi phonetic mapping scheme 600) representing the Hindi words that he or she wishes to appear on the display.

- [40] Referring now to Figure 8a (and in view of Figure 1), a flowchart is presented. The flowchart describes the normal data flow when keyboard input is sent to a computer. In step 801, a key on keyboard 162 is depressed and the keyboard 162 sends input to the computer 110 in the form of a scan code. The scan code is received by computer operating system 134,144 in step 803. In step 805, the operating system 134,144 determines the current keyboard layout and selects the appropriate output based on the keyboard layout. In step 807, the operating system 134,144 sends the Unicode character provided by the combination of the scan code from the keyboard 162 and the keyboard layout that is active in the system to the active application window where it is displayed to the user on an output device such as monitor 191.
- [41] Figure 8b provides a flowchart that shows the data flow that takes place in an embodiment of the present invention. By way of example and not limitation, the input locale of the current active window is set to the Hindi language. However, rather than configuring the keyboard layout to the traditional Hindi keyboard layout, the keyboard layout is configured to use the US English keyboard layout that is widely used in the United States. In step 800, a key on the keyboard 162 (or possibly some other input device known in the art such as a touchpad or stylus on a personal digital assistant) is struck resulting in a scan code being sent to computer 110. In step 802, the scan code is received by the computer operating system 134,144. In step 804, the operating system 134,144 of computer 110 determines the current keyboard mapping and applies the mapping.
- [42] In step 806, the operating system 134,144 determines the active window. Because of the keyboard hook, it may be necessary to make this determination programmatically without relying on the base operating system. One way this may be done is through the use of active accessibility APIs that are well-known in the art. In step 808, the received input is captured by a keyboard hook. A hook is a software component that “hooks” data out of its normal traversal path. In one embodiment, this keyboard hook may be implemented as a Windows public API that is installed to capture all keyboard

input prior to being sent to the active window's (i.e. active application's) input queue. In step 810, the hooked input is transliterated to another alphabet using phonetic mapping engine 502, described above. In step 812, the hooked input is displayed in a preview window to the user in two parts – the English representation of the phonetic string that was typed by the user, and the Hindi result that will be sent to the active application window and displayed to the user. Displaying the inputted data in this way allows the user to see what Hindi string will be displayed in the active application prior to inserting the string into the application. Referring briefly to Figure 9a, an illustrative embodiment of this preview mechanism is shown. An active application window 900 is shown with preexisting text 902. In this example, the user has typed the word 'microsoft' using the US English keyboard layout. The user is provided with a preview of how the word will appear in Hindi by preview popup 904. Within preview popup 904, the English text string 906 of what was typed is displayed, with the corresponding Hindi script 908 displayed immediately above.

- [43] Referring back to Figure 8b, in step 814, a termination character is received by the system, indicating the end of a discrete input string, e.g., the end of a word. This termination character may be a space, a carriage return, or a tab (or some other termination character defined in the system). Once the termination character is received, the system displays the Hindi characters in the active window. As a result, in step 816, the system retrieves the active window (which has been monitored by the active accessibility API), and sends the Unicode characters of the mapped string (Hindi script 908 in the present example) to the active window to be displayed on an output device such as monitor 191.
- [44] In another embodiment of the present invention, the Text Services Framework (TSF) of the Microsoft® Windows operating system may be used to provide additional options for allowing a user to input and transliterate between languages using phonetic mapping engine 502. Interface elements can be added to the language bar 200 that is shown in Figure 2 utilizing TSF. (Interface elements may be added to other

toolbars as well.) Figure 9b provides an example of a language toolbar that has been modified according to aspects of the present invention.

- [45] Modified language bar 910 has language selection columns 912 which are similar to language selection dropdown 402 found in Figure 4. In an embodiment of the invention, an additional column is added to the modified language bar 910 that is used to manage the transliteration that is accomplished by phonetic mapping engine 502. This additional column may be referred to as the transliteration menu 914. The transliteration menu allows a user to transliterate text between any two languages that have phonetic mapping schemes 504 associated with them in the phonetic mapping engine 502. This functionality is possible because the phonetic spelling may be used to bridge the different languages and alphabets together.
- [46] Referring briefly to Figure 10, the flow of converting from a first language to a second language using an intermediary is shown where there is no direct mapping scheme available from the first language to the second language. In this example, transliteration from Hindi text to Telugu text is shown. In step 1000, the Hindi text is sent phonetic mapping engine 502. In step 1002, phonetic mapping engine 502 uses the Hindi phonetic mapping scheme (shown in Figure 6) to map the Hindi characters into a phonetic English language string. Finally, in step 1004, phonetic mapping engine 502 takes the phonetic English language string and converts it to Telugu based on the Telugu phonetic mapping scheme (shown in Figure 7). Thus, by using the English language phonetic mapping as an intermediary that the two languages share in common, the user is able to transliterate Hindi into Telugu.
- [47] Referring back to Fig. 9b, the transliteration menu 914 may have a menu item "Transliterate Selection to" which includes a submenu that contains a submenu entry for each language installed on the system. For example, if English (United States), Kitki, and Hindi are installed on the system, the transliteration submenu may include English, Kitki and Hindi. Transliteration menu 914 may have an additional menu item "Transliterate All" which includes a similar transliteration submenu as is found

in the “Transliterate Selection to” menu item. The “Transliterate All” menu item provides the ability to transliterate all text in the active window from one language to another via the process described above and shown in Figure 10.

[48] Thus, using the above described processes and software applications, a user can input text into a computer using phonetic spellings in a first language’s alphabet, and the computer phonetically maps the input to a second language that uses an alphabet different than the alphabet of the first language. The application running in the background hooks, or captures, all keyboard input and transliterates the input prior to sending the input to the active application, e.g., a word processor, spreadsheet, or any other application into which the user can type. That is, the transliteration is transparent to the active application, and the active application needs no knowledge of the transliteration process. Thus, the invention provides universal transliteration input capability to a user without requiring other software application programs to be modified in any way.

[49] The present invention has been described in terms of preferred and exemplary embodiments thereof. Numerous other embodiments, modifications and variations within the scope and spirit of the appended claims will occur to persons of ordinary skill in the art from a review of this disclosure.